Q3:

For each cell in the matrix, the cost of alignment based on possible operations are computed, minimum of them is chosen.

From the bottom right corner of the matrix to the top-left, the optimal sequence is constructed,

Time complexity analysis

The overall time complexity is $O(m \times n)$ because of the matrix filling step.

Q4:

the maximum discounts are stored in dp-table, it backtracked to determine the maximum discount stores.

The time complexity of this algorithm is $O(n)$ because we iterate the stores.

Q5:

The activation of antenna is made by checking its leftmost coverage, point is to the right of the rightmost coverage point of the last activated antenna, this ~~way~~ algorithm uses greedy approach.

Time comp:

The dominant factor is sorting step which is $O(n\log n)$, Q1:

The 'closest_pair_of_points' function sorts the points based on their x-coordinates and y-coordinates.

The 'closest_pair_helper' function is a recursive function that implements the divide-and-conquer strategy.

The 'min_strip_distance' function checks for points in the strip between the left and right halves that are closer to each other than delta.

The 'brute_force_closest_pair' function is used for small input sizes, where a brute-force approach is more efficient.

The complexity analysis:

the sorting steps take $O(n \log(n))$ time, the recursive parts; $T(n) = 2T(n/2) + O(n)$, this resolves to $O(n \log n)$ using Master, strip part is $O(n)$, so the overall time complexity is $O(n \log n)$

Q2:

The 'min_sensors_to_cover' function sorts the sensors by their x-coordinates and calls the helper.

The helper function recurs by divide and conquer method.

The algorithm considers both halves to find the min number of sensors to create a secure perimeter.

Time complexity Analysis:

Sorting step takes $O(n \log n)$, recursive part is: $T(n) = 2T(n/2) + O(n)$, which is $O(n \log n)$ so the overall time complexity is $O(n \log n)$.